Recuperatorio 1er Parcial

Algoritmos y Estructuras de Datos 3 – DC, FCEyN, UBA

17/11/2021

Para realizar consultas, deben conectarse por Discord al canal de le docente a le cual quieran consultar. Tener en cuenta que une docente no puede conectarse con dos estudiantes en simultáneo. Las aclaraciones de enunciado que podamos llegar a hacer van a ser comunicadas vía Discord al canal de consultas de la práctica.

El examen transcurre de 17:00 a 21:00 hs. A las 21:00 se desconectarán les docentes de sus canales de Discord y tendrán hasta las 23:00 para realizar la entrega vía campus. (Luego de este horario, las entregas serán rechazadas.) El archivo subido al campus puede sobreescribirse una cantidad ilimitada de veces hasta la hora de entrega. Independientemente de si sobreescriben o no, deberán confirmar su entrega definitiva (que ya no podrá sobreescribirse). Sólo en caso de que el Campus estuviera saturado y no funcionara, sería adecuado realizar la entrega por mail a algo3-doc@dc.uba.ar con copia a fsoulign@dc.uba.ar indicando claramente la entrega en el asunto.

El examen puede realizarse **a mano** o en **computadora**. En el primer caso, deben **escanearlo o fotografiarlo** y deben **unir y comprimir** las páginas resultantes para generar un único archivo en **formato PDF** con un **peso razonable**. El resultado debe ser un documento **legible** (buena iluminación, buena resolución, buena orientación, no fotos cortadas, etc.) y que tenga un **orden de lectura claro**, **¡verificarlo!**. En el segundo caso, el formato debe ser **PDF** o **texto plano (txt)**. Por cuestiones de compatibilidad, **no** se aceptan entregas en otros formatos (zip, rar, jpg, gif, png, tiff, etc). **Tampoco** se aceptan entregas de links a repositorios personales.

El examen es personal y pueden usar las teóricas, las clases prácticas y las guías de ejercicios, citando claramente. Las respuestas deben estar debidamente justificadas incluso en aquellos ejercicios en los que este hecho no es recordado.

El examen se **aprueba** con al menos 2 ejercicios aprobados; ver condiciones particulares de aprobación de cada ejercicio.

1) Una biblioteca tiene interés en conseguir n artículos académicos que fueron publicados en m volúmenes de un mismo periódico durante un lapso de k años. Cada artículo se publicó en un único volumen, mientras que cada volumen contiene varios artículos. Asimismo, cada volumen se publicó en un único año en el que se publicaron varios volúmenes. La biblioteca tiene tres opciones para adquirir cada artículo: 1. comprarlo en forma individual, 2. comprar el volumen en que salió publicado el artículo, 3. comprar el compilado de todos los volúmenes que se publicaron el mismo año que el artículo. En el segundo caso la biblioteca adquiere todos los artículos publicados en el volumen, mientras que en el tercer caso la biblioteca adquiere todos los artículos de dicho año.

   Como hay un presupuesto acotado, la biblioteca no puede comprar todos los artículos. Por este motivo, sus bibliotecaries decidieron priorizar los artículos, asignándoles un valor. El objetivo de la biblioteca es maximizar el valor total de los artículos adquiridos (i.e., la suma de los valores individuales) sin superar su presupuesto.

Para cada artículo $1 \leq i \leq n$ se conoce su valor $v_i \in \mathbb{N}$, su volumen e_i $(1 \leq e_i \leq m)$ y su costo $c_i \in \mathbb{N}$ si se compra en forma individual. Para cada volumen $1 \leq j \leq m$ se conoce su costo $p_j \in \mathbb{N}$ y año de publicación a_j. Finalmente, para cada año $1 \leq y \leq k$ se conoce el costo q_y del compilado correspondiente. Para simplificar el ejercicio, suponemos que $e_i \leq i$ y que $a_j \leq j$ (y por ende $k \leq m \leq n$) y que $e_1 \leq e_2 \leq \ldots e_n$ y $a_1 \leq a_2 \leq \ldots \leq a_m$. Es decir, los artículos que fueron publicados en el j-ésimo volumen son consecutivos y los volúmenes que fueron publicados el y-ésimo año son consecutivos.

Ejemplo: si el presupuesto es 11, los artículos valen y cuestan $(4, 5, 3, 2, 4, 1)$, los volúmenes son $(1, 1, 2, 3, 3, 4)$ y cuestan $(7, 4, 4, 6)$, y los compilados son $(1, 1, 2, 3)$ y cuestan $(6, 6, 6)$, entonces el máximo valor posible es 19 y se consigue adquiriendo el compilado del primer año, el tercer volúmen y el sexto artículo.

a) Definir en forma recursiva la función $B\colon \{1, \ldots, n\} \times \mathbb{N} \to \mathbb{N}$ tal que $B(i, q)$ denota el máximo valor total que puede obtener la biblioteca cuando su presupuesto es q, si compra únicamente artículos menores o iguales a i, volúmenes que no tienen artículos mayores a i, y compilados que no tienen artículos posteriores a i.

b) Demostrar que B tiene la propiedad de superposición de subproblemas cuando q es suficientemente chico, explicando cuándo es q suficientemente chico.

c) Definir un algoritmo *top-down* para calcular $B(i, q)$ cuando se obtiene el input indicado previamente, indicando claramente las estructuras de datos utilizadas y la complejidad resultante.

d) Escribir el (pseudo-)código del algoritmo top-down resultante.

Para que el ejercicio se considere aprobado, la función B debe ser correcta y estar bien explicada, mientras que el algoritmo resultante debe tener complejidad $O(nq)$.

2) Diseñar un algoritmo de tiempo $O(n + m)$[1] que, dado un grafo conexo G y un vértice v, determine la cantidad de árboles v-geodésicos distintos de G. Recordar que T es un árbol v-geodésico de G cuando T es un árbol generador de G y la distancia (en cantidad de aristas) entre v y w en T es igual a la distancia entre v y w en G para todo $w \in V(G)$. **Justificar** que el algoritmo propuesto es correcto. **Ayuda:** pensar cuáles aristas pertenecen a árboles v-geodésicos para determinar una fórmula que cuente las posibles formas en que se puede llegar a cada vértice desde la raíz.

   Para que el ejercicio se considere aprobado, la idea del algoritmo debe ser correcta y estar bien explicada, más allá de que la implementación y/o justificación tenga errores.

3) Diseñar un algoritmo de tiempo $O(km)$ que, dado un grafo pesado G cuyos pesos pertenecen a $p_1 < p_2 < \ldots < p_k$, compute un árbol generador mínimo de G. **Justificar** que el algoritmo propuesto es correcto. **Ayuda:** pensar cómo resolver los casos $k = 1$ y $k = 2$ aprovechando el invariante de Kruskal y generalizar para $k > 2$.

   Para que el ejercicio se considere aprobado, el algoritmo debe tener la complejidad pedida y la justificación debe ser convincente.

[1]Puede suponer que las multiplicaciones son operaciones elementales.